Weekly Report(March.4,2019-March.10,2019)

Rui Shaopu

Abstract

This week I learned a lot about FullyNeuralNetwork, Batch Normalization and Dropout. I understand more about neural network through this week's learning.

1 Learned

I learned more about NN, and I get a clearer understanding about it.

1.1 FullyConnected Neural Network

There are more than two layers, but each two contiguous layers are all connected like the two-layers neural network implemented in assignment1. In the hidden layer, as usual we use ReLu as activation function. Output Y is activated by softmax function. By this way, we can also assume it as probability.

Form (L layers):

affine - [batch/layer norm] - ReLu - [dropout] * (L - 1) - affine - softmax

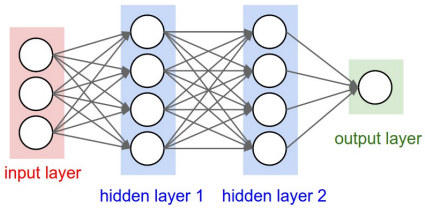

Figure 1: 2-dimension

1.2 Batch Normalization

Advantages:

1.avoid saturation of activation function.

2.avoid too long training time caused by large difference between distribution of training data and test data.

3.use larger learning rate since BN can converge quickly.

Main Formula:

Forward:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\delta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$$

$$\hat{x_i} = \frac{x_i - \mu}{\sqrt{\delta^2 + \epsilon}}$$

$$y_i = \gamma * \hat{x_i} + \beta$$

Backward:

$$\frac{\partial l}{\partial \hat{x_i}} = \frac{\partial l}{\partial \hat{y_i}}\gamma$$

$$\frac{\partial l}{\partial \delta^2} = \frac{-1}{2}\sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}(x_i - \mu)(\delta^2 + \epsilon)^{\frac{-3}{2}}$$

$$\frac{\partial l}{\partial \mu} = \frac{\partial l}{\partial \hat{x_i}}\frac{-1}{\sqrt{\delta^2 + \epsilon}}$$

$$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x_i}}\frac{1}{\sqrt{\delta^2 + \epsilon}} + \frac{\partial l}{\partial \delta^2}\frac{2(x_i - \mu)}{m} + \frac{\partial l}{\partial \mu}\frac{1}{m}$$

simplified: $$\frac{\partial l}{\partial x_i} = \frac{\gamma}{\sqrt{\delta^2 + \epsilon}}[\frac{\partial l}{\partial y_i} - \frac{1}{m}(\hat{x_i}\sum_{j=1}^{m}\frac{\partial l}{\partial y_j}\hat{x_j} + \sum_{i=1}^{m}\frac{\partial l}{\partial y_i})]$$

$$\frac{\partial l}{\partial \gamma} = \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\hat{x_i}$$

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}$$

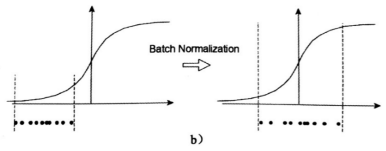

Figure 2: 2-dimension

1.3 Dropout

Like its name, neurals will be dropped temporarily from network according to probability p. By this method, people solved the problem of overfitting and make our network more robust.

It works at the training stage.

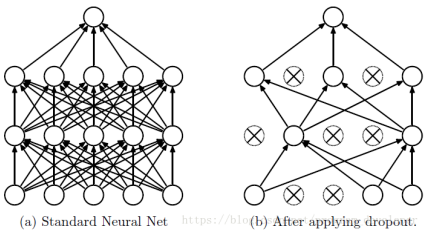

Figure 3: 2-dimension

Without dropout:

$$z_i^{l+1} = W_i^{l+1} y_i^l + b_i^{i+1}$$
$$y_i^{l+1} = f(z_i^{l+1})$$

With dropout:

$$r_i^j \sim Bernoulli(p)$$
$$\tilde{y}_i^l = r^l * y_i^l$$
$$z_i^{l+1} = W_i^{l+1} \tilde{y}_i^l + b_i^{i+1}$$
$$y_i^{l+1} = f(z_i^{l+1})$$

1.4 some results

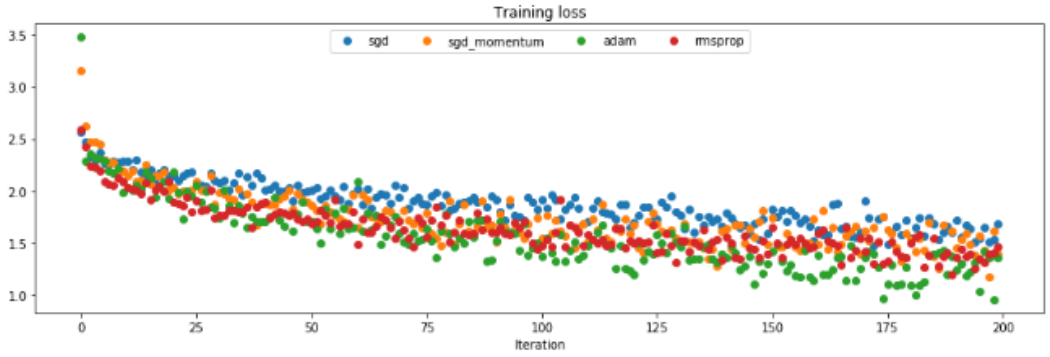

Figure 4: iterations

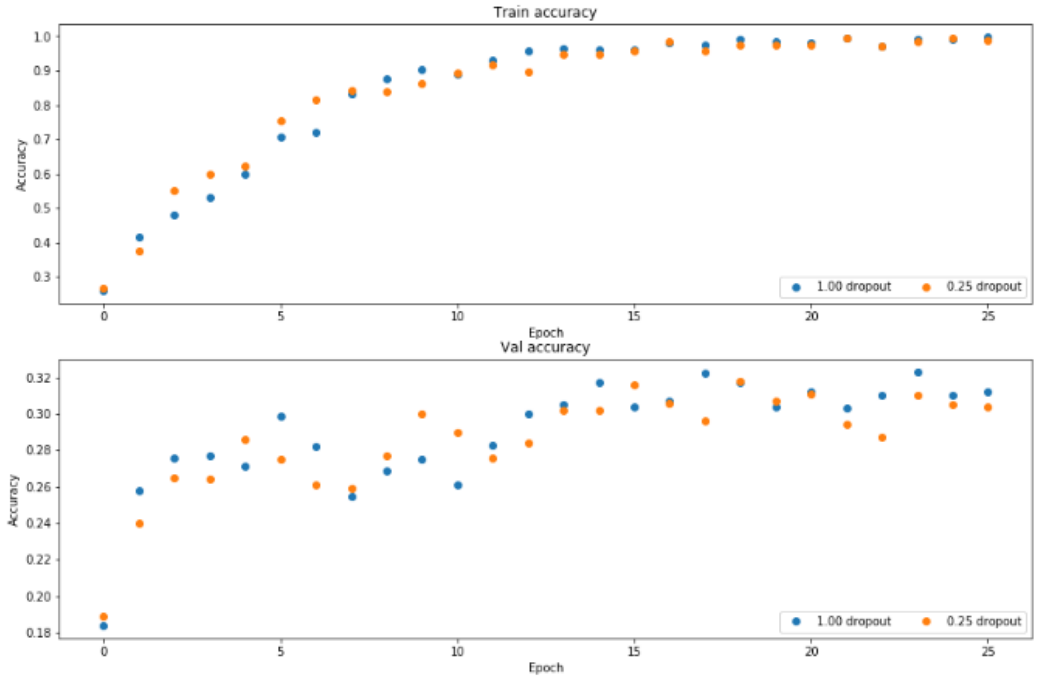

Figure 5: iterations

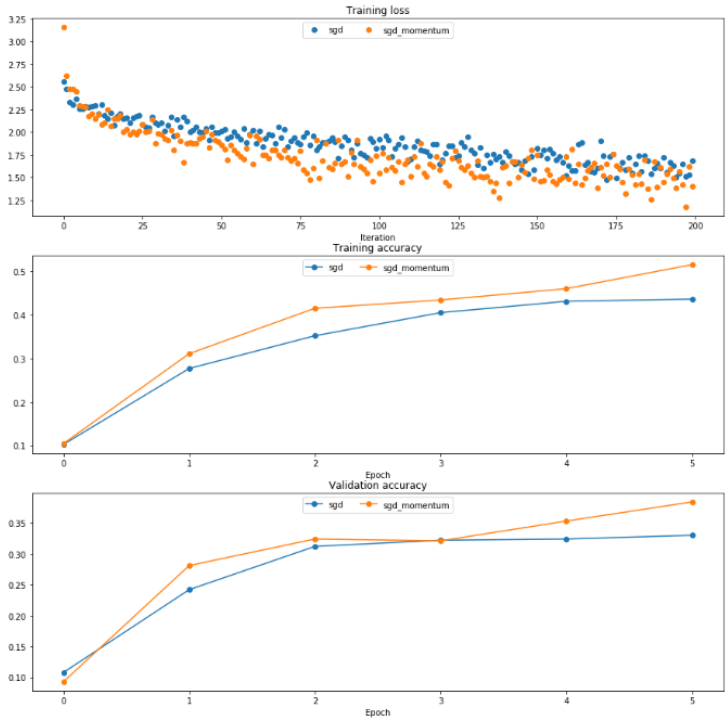

Figure 6: iterations

2 Plan

Next week, I will learn CNN and keep finishing assignment2. Thanks for your reading.